CHAPTER 8

Buses and Interfaces

INDUSTRY EARLY REVIEW DRAFT V.0.3 — 07/20/1999 7:06 PM–

NOTE to REVIEWERS: This is a very early draft version, and no effort has been made to reconcile changes in cross references to other chapters in the guide. Please look for comments such as this in the draft, which encourage your feedback on specific issues.

Please submit comments using the form on http://www.pcdesguide.org or by sending e-mail to comments@pcdesguide.org.

IMPORTANT: The requirements defined in this guide provide guidelines for designing PC systems that will result in an optimal user experience with typical Windows-based applications running under either the Microsoft Windows 98 "Millennium" or later or Windows 2000 Professional or later operating systems. These design guidelines are not the basic system requirements for running any version of Windows operating systems.

Contents

USB

This section presents the requirements for Universal Serial Bus (USB)

USB provides an expandable, hot-pluggable Plug and Play serial interface that ensures a standard, low-cost socket for adding external peripheral devices ranging from interactive HID devices such as joysticks and pointing devices to isochronous devices such as telephony, audio, and imaging devices. USB allows cascading hubs that can be integrated into desktop devices, such as monitors and keyboards.

USB is required on all PC 2001 systems, and migration of I/O devices from legacy ports to USB is recommended. In particular, the joystick, pointing device, and keyboard devices that ship with PC systems should be USB.

Any device that plugs into a USB port is considered a USB device and must comply with the requirements defined in these guidelines. If the device provides the capabilities of one or more functions or it provides a hub to the host, it must comply with the requirements in this chapter.

~~Manufacturers should ensure that their USB devices are tested at the compatibility workshops provided by the USB Implementers Forum.~~

USB ~~Basic~~ Core Requirements

This section covers guidelines for the Universal Serial Bus (USB) specifications 1.1 and 2.0.

[7.1] System includes USB with ~~two~~four USB ports, minimum

Note to Reviewers: Please supply feedback on this requirement.

USB must be included on all ~~PC 99~~PC 2001 system types.

Mobile PC Note

At least ~~two~~four USB ports are required for every system type except Mobile PC, which must include at least one USB port. USB support must be provided for the full bandwidth specified in the *USB Specification, Version 1.~~0~~1* or later. The requirement for four ports can be provided with an external hub if the hub is bundled with the system.

When a system has more than one host controller, each host controller must provide full bandwidth and isochronous support. Host controllers should be located on the PCI bus (or equivalent) to meet this requirement.

[7.2] Systems include BIOS support for USB keyboards and hubs

PC 2001 systems, except those with captive keyboards, such as a mobile PC system, must have BIOS support for USB keyboards and hubs. This support must provide the ability for the user to enter the BIOS setup utility and also

provide enough functionality to install and boot a USB-aware operating system. USB keyboards built as standalone devices, part of a composite device, or part of a compound device must all be recognized and usable. The BIOS is required to support keyboards behind at least one level of external hubs.

For systems with multiple USB host controllers, BIOS support for USB keyboards and hubs is required for ~~at least one of the~~all host controllers that are integrated on the motherboard (that is, not add-on cards).

[7.3] All USB system hardware, hubs, and devices comply with USB ~~1. 0~~1.1 specification

~~Recommended: All USB system hardware complies with USB 2.0 specification.~~

All USB system hardware must comply with *USB Specification, ~~Version 1.0,~~Version 1.1* ~~and should comply with *USB Specification, Version 1.1.*~~. Compliance with the USB specification ensures that USB hardware has complete Plug and Play capabilities and is implemented in a standard way. Compliance with this requirement is demonstrated on the compliance process of the USB Implementers Forum.

~~For example, on any system with USB capabilities, a user must be able to dynamically attach any USB peripheral to any USB connector. The operating system should automatically recognize it, load and initialize the appropriate drivers, and make the device available for use.~~

All USB devices (which includes hubs) must comply with *USB Specification, Version 1.1* or better. Compliance with the USB specification ensures that USB hardware has complete Plug and Play capabilities and is implemented in a standard way. Compliance with this requirement is demonstrated through successful completion of the compliance process of the USB Implementers Forum.

[7.4] [REDUNDANT] Connections use USB icon

Note to Reviewers: This is a basic PC 2001 requirement.

[7.5] USB devices and drivers support maximum flexibility of hardware interface options

Device and driver designs must provide maximum flexibility for interface options so that the operating system or other resource manager can coordinate user preferences, allowing multiple devices and applications simultaneously.

?? 7.5.1. Devices and drivers provide multiple alternate settings. Devices and drivers must provide multiple alternate settings for each interface where any alternate setting consumes isochronous bandwidth.

?? 7.5.2. Devices and drivers must not use isochronous bandwidth for alternate setting 0. Devices should consume bandwidth only when the device is being used.

[7.6] USB host controller meets ~~either~~ OpenHCI, UHCI, or USB 1.1 HC specification

The host controller providing USB1.1 functionality must comply with the specifications for either *Open Host Controller Interface* (OpenHCI), published by Compaq, Microsoft, and National Semiconductor, or *Universal HCI* (UHCI), published by Intel. ~~Hardware manufacturers who design to one of these specifications are not required to provide an additional device driver for their host controller under the Windows or Windows NT Workstation operating systems. Multiple OpenHCI and UHCI USB controllers are supported concurrently by the operating system.~~

[7.7] USB host controller can wake the system

The USB host controller must support wake-up capabilities from S1, S2 and S3 states. ~~S1 or S2. Supporting wake-up from the S3 state is recommended. Notice that if wake-up from the S2 state is supported, wake-up from the S1 state must also be supported. Similarly, if wake-up from the S3 state is supported, wake-up from the S1 and S2 states must be supported.~~

If the system contains multiple USB host controllers, ~~only one is~~ all host controllers integrated on the motherboard (that is, not add-on cards) are required to support wake-up from S1, S2, and S3. ~~capability, although it is recommended that all host controllers support wake-up capability.~~

[7.8] [REDUNDANT] USB hubs comply with USB 1.1 or specifications

Note to Reviewers: This item is now defined in requirement 7.3.

[7.9] All hubs must be self-powered except hubs integrated into USB keyboards ~~Bus-powered USB hubs provide ports that can be individually power switched~~

To minimize USB power consumption requirements, bus-powered hubs must provide ports that can be individually power switched. This contributes to the goal of reducing overall system power consumption. It is especially important in mobile environments, where power consumption must be absolutely controlled when the system is on battery power. Furthermore, if a bus-powered hub is implemented in a USB keyboard, that hub must provide ports that are individually power-switched.

[7.10] Systems and USB devices comply with USB power management requirements

PC 2001 systems and devices must comply with the power management requirements in the *USB Specification, Version ~~1.0~~1.1* or later.

In addition, all devices must comply with the Interface Power Management feature in the *USB Common Class Specification, Revision ~~1.0~~1.1* or later.

[7.11] USB devices meet requirements in related USB device class specification

A USB peripheral that fits into one of the USB device class definitions must comply with the related USB device class specification. USB class drivers in the operating system are implemented to support devices that comply with the particular device class specification.

Class driver extensions and WDM support provided in Windows 98 and Windows 2000 allow IHVs to innovate and differentiate their products while still meeting class compliance in their base operational modes.

Devices can use the generic class drivers provided with the operating system, or manufacturers can create drivers or WDM minidrivers, depending on the device class, to exploit any additional unique hardware features.

[NEW] USB devices install without pre-loading software or drivers and without rebooting the system

For example, on any system with USB capabilities, a user must be able to dynamically attach any USB peripheral to any USB connector. The operating system should automatically recognize it, load and initialize the appropriate drivers, and make the device available for use.

USB Design for Mobile PCs

This section addresses the unique design issues for USB for Mobile PCs.

[6.5] Mobile PC includes at least one USB port

Note to Reviewers: A future draft of the chapter will address issues for when separate host interfaces are present in the mobile platform and in the dock (on dock-side PCI)

For mobile PCs, at least one USB port must be built into the PC, not provided solely by docking stations, although these units can provide extra USB connectors. This USB port can be either a high-power or low-power port, or it can be dynamically configurable at the discretion of the OEM, as provided for by Section 7 of the USB 1.1 specification.

Mobile systems are **not** required to meet the requirement for the USB host controller to be able to wake the system from S3 state, as defined in requirement [3.2], “System design meets ACPI 1.0a specification (or later) and PC 2001 requirements.”

Mobile systems that have built-in keyboards are **not** required to include BIOS support for USB keyboards and hubs as defined in requirement [3.5], “BIOS meets PC 2001 requirements for boot support.”

[6.6] For mobile PC, USB-connected internal device does not maintain fully on power state

An internal device that connects to the Mobile PC using USB must not continually maintain the system in a fully on power state. Such a device will override system power-management settings that control power-saving modes to protect battery life. When any USB device is connected but not active, the driver must allow system power management to suspend the Mobile PC.

IEEE 1394

This section summarizes PC 2001 design requirements for PC platforms designed with one or more integrated IEEE 1394 nodes.

Core Architecture for IEEE 1394 under Windows

Windows 98 and Windows 2000 provide an IEEE 1394 bus driver. A driver for a device that connects to the IEEE 1394 bus sits on top of the IEEE 1394 driver stack. The driver communicates to the device by sending IRPs to the IEEE 1394 bus driver, which the operating system provides as 1394bus.sys. The port driver provides a hardware-independent interface to the IEEE 1394 bus. The port driver handles some IRPs, and ~~some~~ it forwards others to the port driver for the motherboard's host controller. Microsoft provides a standard port driver for host controllers that satisfy the Open Host Controller Interface specification, Ohci1394.sys.

Base-level support for IEEE 1394 audio/video (A/V) devices is provided in Windows 98 and Window 2000 through the WDM Stream class, which supports components such as DVD decoders, MPEG decoders, video decoders, tuners, and audio codecs.

The WDM Stream class supports a uniform model for standard and custom data types, following the kernel streaming conventions described in the Windows 2000 DDK to support data transfer between kernel drivers without requiring a transition to user mode.

Support for IEEE 1394 hard disks, CD-ROM drives, DVD drives, printers, and scanners is implemented through the Serial Bus Protocol (SBP-2) class driver, which communicates through the SBP-2 port driver.

For details about the architecture and driver implementation for devices that use the IEEE 1394 bus, see "Part 6: IEEE 1394 Drivers" in the "Kernel Mode Drivers Design Guide" of the Microsoft Windows 2000 DDK (online at http://www.microsoft.com/DDK/DDKdocs/Win2k/1394-design_17dz.htm).

IEEE 1394 Basic Requirements

The following is a summary of the IEEE 1394 design considerations related to PC systems, as addressed in this chapter:

?? Compliance with IEEE 1394 standards, specifically ~~IEEE 1394-1995 and IEEE P1394.a~~IEEE 1394b-2000

?? Support for the 1394 Open Host Controller Interface (OpenHCI) ~~specification for controllers,~~ specifically OHCI Revision ~~1.0~~ 1.1 and later

?? Plug and Play support for device configuration, control and status registers (CSRs), connectors and cabling, and connection fault handling

?? Cable power distribution, including requirements for source devices, sink devices, self-powered devices, and ~~supporting~~ their applicable CSRs

?? Device power management, CSRs, and soft-power protocols

~~? Device command protocols for audio, video imaging, still imaging, and storage device classes~~

This section defines the basic PC 2001 guidelines for implementing IEEE 1394.

[8.1] Systems integrating IEEE 1394 support mandatory features in IEEE ~~P~~1394~~a~~ b-2000 and provide IEEE 1394-1995 and 1394a-1995 interconnectivity

Designs that interface to the IEEE 1394 bus must support the following industry standards and supplemental specifications:

Note to Reviewers: A future draft will explain the important scope of 1394b-2000 and its relationship to the 1394-1995 spec

?? IEEE 1394b-2000 standard, an amendment to IEEE 1394-1995

?? IEEE 1394a-1999, an amendment to IEEE 1394-1995

?? IEEE 1394-1995 Standard for a High Performance Serial Bus

?? IEEE 1212r-2000

?? ANSI NCITS 3.25-1998 (SBP2)

?? 1394 Trade Association Power Specification (all components)

?? Plug-and-Play Design Specification for IEEE 1394 (http://www.microsoft.com/hwdev/1394/

[8.2] Controllers comply with 1394 OpenHCI 1.1 ~~for IEEE 1394~~

Host controllers must implement the mandatory features of 1394 OHCI Revision 1.1, including support for PCI Power Management (including wake on LinkOn), Dual Buffer Mode enhancements, ack_tardy processing, SCLK failure detection, Skip Processing enhancements, and Block Read Request handling. Optionally, the controller can support out-of-order pipelining enhancements.

Host controllers must support IEEE 1394b-2000 features (for example, B.O.S.S.).

[8.3] [REDUNDANT] OpenHCI controllers and devices support advances defined in IEEE 1394a

[8.4] Host supports peak data rate of ~~400~~ 800 Mb/s, minimum

Note to Reviewers: Please give us input on the level being set to 800.

All host controllers and PHY ports available within the PC system for 2001 must support a peak data rate of 800 Mb/s. The host controller must support 100-Mb/s, 200-Mb/s, and 400-Mb/s data rates as specified in IEEE 1394-1995 and 1394a-1999, 400-Mb/s. and 800 Mb/s per IEEE 1394b-2000.

All externally accessible host controller ports should support bilingual S100-400 operation.

~~A peak data rate of 400 Mb/s is required of all host controllers and PHY ports available externally in the system for 1999. The host controller must support 100-Mb/s, 200-Mb/s, and 400-Mb/s data rates as specified in IEEE 1394-1995 andIEEE P1394.a. All externally-accessible host controller ports must supportS100-400 operation.~~

~~**8.5 [DELETE] Design avoids excessive currents resulting form ground fault potential among devices**~~

Guidelines for IEEE 1394 Devices

This section summarizes additional requirements for IEEE 1394 peripherals.

[8.6] Device command protocols conform to standard device class interfaces

Devices using the SBP2 protocol must conform to the guidelines set in “SBP-2 Support and Windows 2000,” available at http://www.microsoft.com/hwdev/print/sbp2_w2000.htm

~~IEEE 1394 devices must comply with appropriate industry-recognized transport and command standards, such as the following:~~

~~? IEC 61883 parts 1-6, including CIP (Common Isochronous Packet) headers, CMP (Connection management Procedures), and FCP (Function Command Protocol)~~

~~? 1394TA AV/C 3.0 and the AV/C subunit family of specifications~~

~~? National Committee for Information Technology Standards (NCITS) SBP-2 transport protocols~~

~~? National Committee for Information Technology Standards (NCITS) T10, Reduced Block Commands (RBC)~~

~~? National Committee for Information Technology Standards (NCITS) T10 MMC-2, or SFF 8090, Version 3~~

~~Storage-class devices must conform to the ANSI standards for SBP-2 (Serial Bus Protocol) with the appropriate command set: RBC (Reduced Block Commands) or MMC-2 (MultiMedia Commands).~~

~~Printing devices using the SBP2 protocol must conform to the guidelines set in "SBP-2 Support and Windows 2000," available at http://www.microsoft.com/hwdev/print/sbp2_w2000.htm~~

~~Drivers for IEEE 1394 must take advantage of WDM-based driver support provided in the operating system.~~

[8.7] IEEE 1394 devices support peak data rate of ~~400~~ 800 Mb/s, minimum

PC 2001 IEEE 1394b-2000 designs must support 800 Mb/s.

S100 devices are strongly discouraged; and S200 devices should limit their peak bus utilization to less than 50 percent.

A device with more than one port should support S100, S200, and S400 PHY operations. All new peripherals and systems should support S400 and S800 1394b 2000 beta mode. PHY operations at a minimum. Pre-existing devices (for example, devices designed prior to the publication of these guidelines) supporting only S100 or S200 PHY ports are acceptable.

Note to Reviewers: Please supply feedback on the 800 Mb/s minimum.

Plug and Play for IEEE 1394

This section summarizes the Plug and Play requirements for IEEE 1394 peripheral devices and PC host controllers.

[8.9] IEEE 1394 Plug and Play devices demonstrate interoperability with other devices

All devices must support Plug and Play for intended applications in both a minimal and an extended bus configuration. A minimal configuration is the minimum number of devices necessary to demonstrate the primary application of the device. An extended configuration is an advanced application with at least two devices added to the minimal configuration. The added devices can be extraneous to the application.

~~**8.10 [DELETE] Topology faults do not cause the bus to fail**~~

[8.11] [REDUNDANT] Removable media devices support media status notification

Note to Reviewers: See the related guidelines in the Storage chapter

[8.12] IEEE 1394 devices that initiate peer-to-peer communications provide a remote control interface

All devices capable of initiating peer-to-peer communications that have been designed for use with the PC must provide a remote interface (enabling remote control for PC applications) that allows a third device, such as a PC or some other device controller, to initiate data transmission between two devices.

Plug and Play for Configuration ROM

This section defines the Plug and Play requirements related to configuration ROM.

[8.13] IEEE 1394 Configuration ROM is provided for unique device identification

The device configuration ROM must provide configuration information as specified in the IEEE 1212r-2000 standard and applicable IEEE 1394 standards, thus providing Plug and Play device control.

[8.14] IEEE 1394 device configuration ROM implements general ROM format

The general configuration ROM format is specified in the applicable IEEE 1394 standards and the IEEE 1212r-2000 standard. The general ROM format is an extensible tree structure enabling a managed environment by providing node-specific and unit-specific information as required for Plug and Play, power management, and isochronous data transfers. The general ROM format also provides for definition of multifunction device units. The bus information block and root directory of the general ROM format are required as specified in configuration ROM table.

~~**8.15 [DELETE] Bus information block implemented at a base address offset of 0404h**~~

~~**8.16 [DELETE] Configuration ROM provides globally unique device ID**~~

~~**8.17 [DELETE] Root directory is located at a fixed address following the bus information block**~~

[8.18] IEEE 1394 configuration ROM includes a unit directory for each independent device function

A unit directory is required for independent function and control of each device unit. A valid pointer to a unit directory must be provided in the root directory. ~~at~~

~~offset 0x20h, in compliance with the general ROM format specified in IEEE 1394-1995 and the directory format specified in ISO/IEC 13213:1994.~~

~~8.19 [DELETE] Each unit directory provides a valid Unit_Spec_Id and Unit_Sw_Version~~

~~8.20 [DELETE] Each unit directory provides a pointer to a unit dependent directory~~

[8.21] Vendor and model leafs support textual descriptor leaf format

Textual descriptors are required for ~~Unit_Spec_ID and Unit_Sw_Version~~Vendor_ID and Model_ID entries in the configuration ROM in order to display this information to the user.~~Textual descriptors are recommended for all other configuration ROM entries.~~ Each textual descriptor points to a leaf that contains a single character string.

Examples of valid textual descriptors are found in the 1394 Plug and Play specification.

~~8.22 [DELETE] Unit dependent directory provides a pointer to the unit's CSRs~~

[8.23] Desktop system provides more than one connector port

The PC must provide at least two externally accessible ports. A Mobile PC must provide at least one externally accessible port.

[8.24] System uses IEEE 1394b-2000 supported sockets

The PC must provide external interconnect to legacy nodes (for example, interconnect to 1394-1995 and/or 1394a-1999 devices).

Note: There must not be a mixture of IEEE 1394 sockets on the back panel of PC2 001 platform implementations.

The connector described in the *Device Bay Specification, Version1.0* (online at http://www.device-bay.org/) is for use inside the PC 2001 platform and is not, in general, considered to be an externally available socket.

~~8.25 - 8.35 [DELETE]~~

Power Management for IEEE 1394 Devices

All devices on the IEEE 1394 bus must comply with the power management requirements outlined in this section.

[8.36] Power Manager is notified of device power state changes

The host controller and all devices that provide or consume cable power must conform to all components of the *1394 Trade Association Power Specification*.

[8.37] Devices and controllers comply with all components of the 1394 Trade Association Power Specification

The Power Specification has been defined to provide guidelines for implementation of devices that propagate, source, or sink cable power. In addition, mechanisms are defined by which devices consuming cable power may be enabled as well as placed into a variety of power consuming states.

[8.38] [REDUNDANT] Devices and controllers comply with 1394 power specification

SCSI

This section presents guidelines for the small computer system interface (SCSI), a flexible I/O bus that is used in the design of a wide variety of peripherals, including disk drives, CD drives, tape drives, scanners, and magneto-optical drives. The SCSI host adapter is the circuitry that serves as an interface between the system and one or more SCSI peripherals. A host adapter can be a card that plugs into the system's expansion bus, such as a PCI card, or it can be designed directly into the system board.

SCSI Host Adapter Requirements

This subsection summarizes class specifications and standards for SCSI host adapters.

[NEW] SCSI controllers comply with SPI-3 standards

All SCSI controllers must meet the hardware and software design requirements listed in the SCSI Parallel Interface 3 (SPI-3) or later standard.

[11.1] SCSI host controller supports bus mastering and virtual DMA services

The host controller must support PCI bus mastering; PCI bus mastering must be enabled by default and virtual DMA services must be supported in the host-adapter option ROM.

Note to Reviewers: Redundant references to ROMs supporting Bus Mastering and Virtual DMA services are combined here. Other locations in the document will be deleted.

[11.2] Bootable SCSI controller supports El Torito No Emulation mode

SCSI host adapters with boot ROMs running in ~~x86~~Intel Architecture platforms must support the current No Emulation mode of the *Bootable CD-ROM Format Specification, Version 1.0 (El Torito)*, or the *BIOS Boot Specification, Version 1.01*.

Note to Reviewers: This was changed for clarification.

[11.3] Option ROM supports Int 13h Extensions

Details are defined in requirement 3.5 in Chapter X, "PC 2001 Core System Guidelines"

[11.5] Bus type is clearly indicated on connectors for all adapters, peripherals, cables, and terminators

Connectors must comply with the requirements defined in the SCSI-2 or later standard. The SCSI bus cable must be plugged into shrouded and keyed connectors on the host adapter and devices. For internal configurations, Pin 1 orientation must be designated on one edge of the ribbon cable and also on the keyed connector of the SCSI peripheral device. For more information, see requirement 3.18, "Connections use icons, plus keyed or shrouded connectors, with color coding."

Although an external connector is optional, if an external connector is provided, it must be a high-density connector as defined in the SCSI-2 or later standard.

Connectors for each SCSI adapter, peripheral, cable, and terminator must be clearly labeled to indicate the bus type. All external SCSI connectors must display the appropriate SCSI icon defined in *Small Computer Interface Parallel Interface* (SPI) standard, Annex H, and must display any clarifying abbreviations or acronyms. The following are applicable acronyms and their definitions:

?? **DIFF (differential).** A signaling method that employs differential drivers and receivers to improve signal-to-noise ratios and increase maximum cable lengths. This method includes both low voltage differential (LVD) and high voltage differential (HVD) types.

?? **SE (single-ended).** A signaling method that employs drivers and receivers to increase circuit density.

?? **LVD (low voltage differential).** A signaling method similar to DIFF but with lower signaling voltages supporting higher transfer rates.

?? **HVD (high voltage differential).** A signaling method similar to DIFF but with higher signaling voltages.

[11.6] Differential Devices support DIFFSENS as defined in the SPI-3 Standard

Without DIFFSENS, the differential bus drivers, a single-ended device, or both could be damaged if a single-ended device is connected to a differential bus.

The standard for DIFFSENS is defined in Section 5.4.2 of the SPI-3 standards document.

[11.7] Automatic termination circuit and SCSI terminators meet SCSI-3 standard

SCSI add-on adapters and on-board controllers must use automatic termination, which allows a user to add external devices without removing the PC case. Terminators used in the SCSI host adapter must be regulated terminators, also known as active, SCSI-3 SPI, SCSI-2 alternative-2, or Boulay terminators. SCSI termination built onto internal cables must meet SCSI-3 standard.

Note to Reviewers: Feedback is required on this topic with regards to Auto Termination circuits and their potential to for too much capacitive loading. For a specific installation is there a solution for user to not have to open the case, yet still have access to the termination circuits through SW or switches? Is this a non-problem?

[11.8] Terminator power is supplied to the SCSI bus with overcurrent protection

This requirement has two components:

?? **[11.8.1] Host adapter must supply terminator power.** The base requirement for system-board implementations using PCI or another expansion bus is that the host adapter must supply terminator power (TERMPWR) to the SCSI bus. All terminators on the host adapter, as well as those on the internal and external SCSI bus, must be powered from the TERMPWR lines on the SCSI bus.

[11.8.2] The circuit that supplies TERMPWR must have built-in overcurrent protection. Devices that provide TERMPWR must also provide some means of limiting the current through use of a self-resetting device. For example, a positive-temperature coefficient device or circuit breaker can be designed into the circuit. These devices open during an overcurrent condition and close after the end of the over-current condition.

Mobile PC Note

Although ~~recommended,~~encouraged, this ~~item~~ feature is not required for battery-powered systems that implement the SCSI host adapter as a PC Card device because of battery consumption issues.

Note to Reviewers: 11.9 and 11.11 were removed and consolidated in 11.5.
11.12 through 11.14 were moved to the Storage chapter to cover devices.
11.15 was removed due to application at a general host bus level. Also, since SCAM is not required, device application was no longer applicable.
11.17 was removed as redundant.
Other items moved to the Storage chapter to eliminate redundancy.

[11.10] Controller and peripherals implement SCSI bus data protection signal

The SCSI host adapter and all SCSI peripherals must implement the SCSI bus data protection signal defined in the SPI standard, and data protection must be enabled by default. This signal was formerly referred to as the parity signal.

Plug and Play for SCSI Host Adapters

This subsection summarizes the Plug and Play requirements for SCSI controllers.

[11.16] [REDUNDANT] Dynamic resource configuration is supported for all controllers

Note to Reviewers: This is a basic PC 2001 requirement that is no longer cited in every chapter

[11.18] SCAM support is disabled by default

~~SCSI Configured Automatically (SCAM) support is not recommended.~~ If support is present, it must be disabled by default. SCAM is not supported by Windows operating systems; enabling SCAM can cause the system to become unstable or inoperable.

[11.20] SCSI controllers provide multi-initiator support

Multi-initiator support allows two SCSI controllers—each installed in a separate computer system—to coexist on a shared SCSI bus with a set of shared devices. If this capability is supported, the SCSI IDs must be changeable from the default SCSI controller ID of 7 and the boot-time SCSI bus reset operation must be able to be disabled on each controller attached to a shared bus.

This capability is recommended for hardware that will be used on systems using the clustering service available under Microsoft Windows 2000 Advanced Server. To use this service, a SCSI adapter and a SCSI peripheral must provide multi-initiator support for at least two initiators.

ATA and ATAPI

This section presents the requirements for Windows-compatible ATA (AT Attachment), ATAPI (ATA Packet Interface) controllers and peripherals.

ATA—also known as IDE (Integrated Device Electronics)—is one of the most widely used interfaces in the PC world.

The use of ATA in a PC 2001 system is optional. If ATA is used, however, all components must comply with the requirements defined in this chapter.

[10.1] IDE controllers comply with ATA/ATAPI-4 standards

All ATA/ATAPI controllers must meet the hardware and software design requirements listed in the ATA/ATAPI-4 or later standard.

Note: External storage subsystems that require advanced features such as command queuing should use IEEE 1394 for the storage interface. Internal primary storage that requires these advanced features should use SCSI.

[10.2] Bootable ATA controller supports El Torito No Emulation mode

Details are defined in requirement 3.5 in Chapter X, "PC 2001 Core System Guidelines"

10.3 [REDUNDANT] Option ROMs support Int 13h Extensions

Note to Reviewers: This basic PC 2001 requirement is defined in 3.5

[10.4] If implemented, dual ATA adapters use single FIFO with asynchronous access or dual FIFOs and channels

Although the use of an ATA adapter with more than one channel is optional, if included, dual ATA adapters must be designed so that either channel might be used at any time; the operating system does not have to serialize access between the primary and secondary channel. This requirement means either that the two channels are totally independent or that anything shared, such as a programmed I/O (PIO) read pre-fetch buffer, is protected by a hardware arbitrator.

Section 5.0 of the *Compaq, Intel, and Phoenix BIOS Boot Specification* defines an implementation for dual asynchronous channels. This specification is available at http://www.ptltd.com/techs/specs.html.

A design implementing a single first in/first out (FIFO) that uses a hardware solution to synchronize access to both channels meets this requirement. ~~If the design does not require that~~A request on one channel need not be completed before another request to the other channel can be started. A software-based solution is not acceptable.

Note to Reviewers: 10.5 requirement was combined with section 10.3 above.

[10.6] [REDUNDANT] System BIOS supports ARMD

Note to Reviewers: This guideline is defined in requirement 3.5.

[10.7] Controller supports Ultra DMA (ATA/33 or higher)

The programming register set for PCI IDE bus master direct memory access (DMA) is defined in ATA-4. ATA drives must comply with ATA-4 to ensure fully featured hardware and Windows-compatible device driver support.

All controllers and ATA hard drive peripherals must support Ultra DMA at transfer rates of 33 MB per second or higher as defined in ATA/ATAPI-4. In addition to improved transfer rates, Ultra DMA also provides error checking for improved robustness over previous ATA implementations. PCI chip sets must implement DMA as defined in ATA-4.

The system BIOS should configure the drive and host controller, optimized for Ultra DMA operation if possible. However, programmed I/O (PIO) mode must continue to work. The ACPI software should also support the restoration of these settings in ACPI control methods _GTM, _STM, and _GTF, for which there are no standard registers, if the controller loses timing context across a suspend and resume cycle. The BIOS pre-operating system boot disk services, (INT13h read and write) need not actually use Ultra DMA for access of the drive prior to operating system boot.

Definitions for the above ACPI control methods can be found in Section 5 of the *Advanced Configuration and Power Interface Specification, Revision 1.0* or later, with consideration of the ACPI errata available on the web site at http://www.teleport.com/~acpi/tech.htm.

[10.8] Controller and peripheral connections include Pin 1 cable designation with keyed and shrouded connectors

Pin 1 orientation must be designated by one edge of the keyed ribbon cable and also on the keyed connector of the ATA or ATAPI controller and peripheral device. Designation of the keyed connector must be clearly indicated on or near the connector.

[10.11] BIOS enumeration of all ATAPI devices complies with ATA/ATAPI-4

The ATA/ATAPI-4 standard defines the enumeration process for all ATAPI devices.

Note to Reviewers: 10.11 will move to the system BIOS requirements item 3.5 in the next draft.
10.9, 10.10 and 10.12 have been moved to the storage chapters.

[10.13] [REDUNDANT] Each controller has a Plug and Play device ID

Note to Reviewers: This basic PC 2001 requirement is no longer defined in each chapter.

[10.14] [REDUNDANT] Dynamic resource configuration is supported for all controllers

[10.15] [REDUNDANT] Resource configuration meets bus requirements

Note to Reviewers: Other items removed as out-of-date.

[10.17] ATA Channel complies with device class power management reference specification

The ATA channel must comply with the *Storage Device Class Power Management Reference Specification, Version 1.0a* or later.

Note to Reviewers: 10.18 was moved to the storage chapter.

PCI

This section presents the PC 2001 guidelines for Peripheral Component Interconnect (PCI) host controllers and peripherals.

The PCI architecture has become the most common method used to extend PCs for add-on adapters. Windows 98 and Windows 2000 use the basic PCI infrastructure to gain information about devices attached to the PCI bus. The ability of PCI to supply such information makes it an integral part of the Plug and Play architecture in Windows.

PCI Basic Guidelines

This section summarizes the basic design guidelines for PCI.

[9.1] All PCI components comply with PCI 2.2

All cards, bridges, and devices that use PCI must be designed to meet the requirements defined in *PCI Local Bus Specification, Revision 2.2* (PCI 2.2). Compliance with this requirement is demonstrated based on the compliance process of the PCI SIG.

[9.3] PCI-to-PCI bridges comply with the *PCI to PCI Bridge Specification, Revision 1.1*

In particular, non-subtractive decode PCI bridges must implement the standard method to close BAR windows. Setting the base address register (BAR) to its

maximum value and the limit register to zeros should effectively close the I/O or memory window references in that bridge BAR.

[9.4] System provides 3.3 V to all PCI connectors

PC 2001 systems are required to provide 3.3 volts with amperage as defined by PCI 2.2 to all PCI connectors. This requirement enables the development of 3.3 V PCI adapters without the cost of voltage regulators.

[9.5] PCI add-on devices support 3.3 V signaling

PCI add-on devices may optionally be Universal Boards as defined in Section 4.1 of the PCI 2.2 specification, or be 3.3 V devices that are "5 V tolerant." These devices support both 3.3 V and 5 V signaling.

PCI Controller Guidelines

This section summarizes PCI controller requirements.

[9.6] System-board bus complies with PCI 2.2

The system-board bus hardware must comply with PCI 2.2. The bus design must fully implement all bus requirements on every expansion card connector.

[9.9] All PCI devices complete memory write transaction (as a target) within specified times

All devices must comply with the Maximum Completion Time requirements that are documented in PCI 2.2. Complying with this requirement ensures shorter transaction latencies on PCI, allowing more robust handling of isochronous streams in the system.

Plug and Play for PCI Controllers and Peripherals

This section summarizes the Plug and Play requirements for PCI devices.

[9.11] PCI Device IDs include Subsystem IDs

The Subsystem ID (SID) and Subsystem Vendor ID (SVID) fields must comply with the Subsystem ID requirement in PCI 2.2. See the white paper "PCI Device Subsystem IDs for Windows," available at http://www.microsoft.com/hwdev/devdes/pciids.htm.

?? The PCI SIG assigns valid, non-zero Vendor ID values to member companies. This Vendor ID value must be used to populate the SVID register.

?? The vendor assigns values for the SID register. To be valid, these values must be non-zero and unique to a subsystem configuration.

Note to Reviewers: The following material was published in PC99A and presented again here to ensure review and to gather industry input on this section. This material will not be included in the final draft.

Valid non-zero values in the SVID and SID registers are necessary for the correct enumeration of the PCI device. When these registers are populated correctly for a PCI subsystem or add-on board, the operating system can differentiate between subsystems and add-on boards based on the same PCI chip.

The PCI specification and these guidelines require that the SVID and SID registers are loaded with valid non-zero values before the operating system accesses the Configuration Space registers on a PCI device or function. This action is required both at initial operating system load and after any transition of the PCI bus from B3 (the unpowered bus state) back to B0 (the fully powered bus state).

For add-on boards, this requirement must be done by hardware on the board itself—for example, by way of serial EEPROM—and not by an extension BIOS or device driver. This action is required because the extension BIOS code or driver code is not guaranteed to run in all relevant cases, especially for system sleep transitions or dynamic bus power state transitions in which the bus becomes unpowered. Hardware methods to support this include:

?? Pin strapping at Reset

?? Loading from an attached parallel or serial ROM

For subsystems on system boards that contain a PCI device, the SVID and SID registers must also be loaded with valid non-zero values before the operating system accesses the device. The exceptions to this requirement are PCI-to-PCI bridges and core chip sets.

For subsystems on system boards that contain a PCI device, the SVID and SID registers must also be loaded with valid non-zero values before the operating system accesses the device. The subsystem exceptions to this requirement are certain sub-classes of bridges and core chipset components, which are specified in section 6.2.4 and Appendix D of the *PCI 2.2 Local Bus Specification*. The PCI 2.2 specification became the industry standard on December 18, 1998. For the convenience of the reader, the excepted sub-classes of bridges (PCI base class 6) and core chip set components (PCI base class 8) are listed here, but for full information the reader must refer to the PCI 2.2 specification:

?? Bridges (PCI base class 6)
  ?? Host bridge (Sub-class 0)
  ?? Host bridge (Sub-class 0)
  ?? ISA bridge (Sub-class 1)
  ?? EISA bridge (Sub-class 2)
  ?? MCA bridge (Sub-class 3)
  ?? PCI-to-PCI bridge and Subtractive Decode PCI-to-PCI bridge (Sub-class 4)
?? Core chip set components (PCI base class 8)
  ?? Generic 8259, ISA, EISA, and I/O APIC programmable interrupt controllers (Sub-class 0)
  ?? Generic 8237, ISA, and EISA DMA controllers (Sub-class 1)
  ?? Generic 8254, ISA, and EISA system timers (Sub-class 2)
  ?? Generic and ISA RTC controllers (Sub-class 3)

Audio/modem riser (AMR) devices and modem riser (MR) devices on the motherboard are not exempt from the requirement for SID and SVID.

The system BIOS power-on self test (POST) code or ACPI control methods (_PS0 for PCI bus B3 to B0 transitions) are guaranteed to run before the operating system accesses the SVID or SID registers. Once the operating system has control of the system, the SVID and SID registers must not be directly writable—that is, the read-only bit must be set and valid. See the note on using the POST method for loading SVID and SID register values related to multiple-monitor support for display devices in requirement [14.45], “Each device has a Plug and Play device ID.”

[9.13] PCI interrupt routing is supported using ACPI

The system must provide interrupt routing information using a _PRT object, as defined in Section 6.2.3 of the ACPI 1.0b specification.

[9.14] BIOS does not configure I/O systems to share PCI interrupts

This applies to boot devices configured by the BIOS on systems based on Intel Architecture processors. The operating system must configure all other devices. For systems that will run the Microsoft Windows family of operating systems, OEMs must design the BIOS so that it does not configure the I/O systems in the PC to share PCI interrupts for boot devices.

An exception exists for legacy audio devices following the configuration guidelines outlined in *Implementing Legacy Audio Devices on the PCI Bus*, available at http://www.intel.com/pc-supp/platform/ac97/wp/leg_pci.htm.

Windows does not support sharing an IRQ between real-mode and protected-mode code within the I/O subsystem. For ~~An~~ example ~~of this is~~, when an NDIS 2.0 driver (real mode) and a SCSI miniport driver (protected mode) for two PCI devices ~~that~~share the same IRQ~~. The problem is that~~, the IRQ needs to be reflected to real mode for the NDIS 2.0 driver to work.

However, if the IRQ is reflected to real mode, the real-mode SCSI driver, which usually is not called because Windows takes over in protected mode, might touch the hardware, causing the SCSI miniport to be confused. Windows resolves this problem either by switching everything to protected mode or by falling back to real mode.

[9.15] BIOS configures boot device IRQ and writes to the interrupt line register

This requirement applies to boot devices configured by the BIOS on systems based on Intel Architecture processors. Windows should configure all other devices because, after an IRQ is assigned by the system BIOS, Windows cannot change the IRQ. If the BIOS assigns the IRQ and Windows needs it for another device, a sharing problem occurs.

The BIOS must configure the boot device IRQ to a PCI-based IRQ and must write the IRQ into the interrupt line register 3Ch, even if the BIOS does not enable the device. This way, the operating system can still enable the device with the known IRQ at configuration time, if possible.

[9.16] System that supports hot plugging for any PCI device uses ACPI-based methods

Hot-plugging capabilities are not required for PCI devices. Windows 98 and Windows 2000 support dynamic enumeration, installation, and removal of PCI devices only if there is a supported hardware insert/remove notification mechanism.

The appropriate notification mechanism is supported as a bus standard for CardBus bus controllers. For other solutions, such as those required for docking stations or hot-plugging PCI devices, the hardware insert/remove notification mechanism must be implemented as defined in Section 5.6.3 of the ACPI 1.0a specification (or later).

[NEW] System that supports PCI-X complies with PCI-X 1.0

All cards, bridges, and devices that use PCI-X must be designed to meet the requirements defined in *PCI-X Specification, Revision 1.0* (PCI-X 1.0). Four-slot, 660 MHz desktop PC are expected in the marketplace by mid-2001.

Power Management for PCI Controllers and Peripherals

This section summarizes the specific PCI power management requirements.

[9.17] All PCI components comply with PCI Bus Power Management Interface specification

PCI Bus Power Management Interface Specification, Revision 1.1 or later, is the only industry specification that ensures compatibility with the power management capabilities of Windows 2000, which uses PME# as the wake-up signal.

The PCI bus, any PCI-to-PCI bridges on the bus, and all add-on capable devices on the PCI bus must comply with the *PCI Bus Power Management Interface Specification, Revision 1.1* or later, whether or not the system they are installed in provides 3.3Vaux to its PCI connectors.

A method that PCI add-on cards can use to meet this requirement is described in Section 7.4.4 of the *PCI Bus Power Management Interface Specification*: Static FET switches on the add-on card re-route 3.3VPCI, or converted 5VPCI, to the 3.3Vaux power plane when the card is installed in a system that does not supply 3.3Vaux. However, the added cost of these static FET switches is not justified for PCI add-on cards installed exclusively in OEM systems.

On OEM systems that supply 3.3Vaux, the OEM-version PCI add-on card's split Vaux power plane is tied directly to the 3.3Vaux pin on the system PCI connector. For systems that do not deliver 3.3Vaux, the OEM-version PCI add-on card's Vaux power plane is tied directly to the required 3.3VPCI source.

PC add-on cards designed and built exclusively for installation in OEM systems—and which are never sold through retail distribution channels—are not required to supply the static FET switches described in section 7.4.4 of the *PCI Bus Power Management Specification*.

This PC 2001 requirement includes correct implementation of the PCI Configuration Space registers used by power management operations, and the appropriate device state (Dx) definitions.

Functions (for example, PCI-to-PCI bridges, USB host controllers, IDE controllers, and so on) that are integrated as part of the core chipset, and thus not add-on capable devices, can use ACPI (and not PCI Power Management registers) for their power management interface.

[9.18] System provide s support for 3.3 Vaux ~~if a system supports S3 or S4 states~~

System support for delivery of 3.3 Vaux to the PCI bus must be capable of powering a single PCI slot with 375 mA at 3.3 V and it must also be capable of powering each of the other PCI slots on the segment with 20 mA at 3.3 V whenever the PCI bus is in the B3 state.

Note to Reviewers: Mobile vendors—provide feedback on whether this requirement is too high for Mobile systems.

Systems must be capable of delivering 375 mA at 3.3 V to all PCI slots whenever the PCI bus is in any "bus powered" state: B0, B1, or B2.

[9.19] PCI bus power states are correctly implemented

Note to Reviewers: This is redundant because it is in the ACPI specification. Our goal is to not reproduce spec data in these guidelines.

The PCI bus must be in a bus state (B*x*) no higher than the system sleeping state (S*x*). This means that if the system enters S1, the bus must be in B1, B2, or B3. If the system enters S2, the bus must be in B2 or B3, and if the system enters S3, the bus must be in B3. Of course, in S4 and S5, the system power is removed, so the bus state is B3. A PCI bus segment must not transition to the B3 state until all downstream devices have transitioned to D3.

Control of a PCI bus segment's power is managed using the originating bus bridge for that PCI bus segment.

?? For CPU-to-PCI bridges, these controls must be implemented using ACPI or the PCI Power Management Interface Specification, Revision 1.1 or later.

?? For PCI-to-PCI bridges, these controls must be implemented in compliance with the PCI Power Management Interface Specification, Revision 1.1 or later.

[9.20] PCI-based modem and network adapters support wake-up

PCI-based modem and network adapters must support wake-up as follows:

?? Modem adapters must be capable of generating a power management event (PME# assertion) from the D3 cold device state. ~~It is recommended that~~ Modem adapters should support capture of Caller ID with hardware support for the AT+VRID, "resend caller ID," voice modem command.

?? Network adapters must support the generation of a power management event (PME# assertion) from the D3 cold device state if the physical layer technology is generally capable of operating under the voltage and current constraints of the D3 cold device state. Network adapters must also support the minimum requirements for network packet filtering/wake-up capability as defined in the requirement 20.56, "Device supports wake-up events."

Mobile PC PCI Issues

The following guideline describes a PCI issue on a Mobile PC.

[6.26] System supports PCI docking through a bridge connector

See complete details in Chapter X, "Mobile PC 2001."

[13.8] [REDUNDANT] All devices meet PC 2001 general device requirements

Note to Reviewers: This is a basic PC 2001 requirement that is not repeated in every chapter

Checklist for Buses and Interfaces

[7.1] System includes USB with four USB ports, minimum

[7.2] Systems include BIOS support for USB keyboards and hubs

[7.3] All USB system hardware, hubs, and devices comply with USB 1.1 specification

[7.4] [REDUNDANT] Connections use USB icon

[7.5] USB devices and drivers support maximum flexibility of hardware interface options

[7.6] USB host controller meetsOpenHCI, UHCI, or USB 1.1 HC specification

[7.7] USB host controller can wake the system

[7.8] [REDUNDANT] USB hubs comply with USB 1.1 or specifications

[7.9] All hubs must be self-powered except hubs integrated into USB keyboards

[7.10] Systems and USB devices comply with USB power management requirements

[7.11] USB devices meet requirements in related USB device class specification

[NEW] USB devices install without pre-loading software or drivers and without rebooting the system

[6.5] Mobile PC includes at least one USB port

[6.6] For mobile PC, USB-connected internal device does not maintain fully on power state

[8.1] Systems integrating IEEE 1394 support mandatory features in IEEE 1394b-2000 and provide IEEE 1394-1995 and 1394a-1995 interconnectivity

[8.2] Controllers comply with 1394 OpenHCI 1.1

[8.3] [REDUNDANT] OpenHCI controllers and devices support advances defined in IEEE 1394a

[8.4] Host supports peak data rate of 800 Mb/s, minimum

[8.6] Device command protocols conform to standard device class interfaces

[8.7] IEEE 1394 devices support peak data rate of 800 Mb/s, minimum

[8.9] IEEE 1394 Plug and Play devices demonstrate interoperability with other devices

[8.11] [REDUNDANT] Removable media devices support media status notification

[8.12] IEEE 1394 devices that initiate peer-to-peer communications provide a remote control interface

[8.13] IEEE 1394 Configuration ROM is provided for unique device identification

[8.14] IEEE 1394 device configuration ROM implements general ROM format

[8.18] IEEE 1394 configuration ROM includes a unit directory for each independent device function

[8.21] Vendor and model leafs support textual descriptor leaf format

[8.23] Desktop system provides more than one connector port

[8.24] System uses IEEE 1394b-2000 supported sockets

[8.36] Power Manager is notified of device power state changes

[8.37] Devices and controllers comply with all components of the 1394 Trade Association Power Specification

[8.38] [REDUNDANT] Devices and controllers comply with 1394 power specification

[NEW] SCSI controllers comply with SPI-3 standards

[11.1] SCSI host controller supports bus mastering and virtual DMA services

[11.2] Bootable SCSI controller supports El Torito No Emulation mode

[11.3] Option ROM supports Int 13h Extensions

[11.5] Bus type is clearly indicated on connectors for all adapters, peripherals, cables, and terminators

[11.6] Differential Devices support DIFFSENS as defined in the SPI-3 Standard

[11.7] Automatic termination circuit and SCSI terminators meet SCSI-3 standard

[11.8] Terminator power is supplied to the SCSI bus with overcurrent protection

[11.10] Controller and peripherals implement SCSI bus data protection signal

[11.16] [REDUNDANT] Dynamic resource configuration is supported for all controllers

[11.18] SCAM support is disabled by default

[11.20] SCSI controllers provide multi-initiator support

[10.1] IDE controllers comply with ATA/ATAPI-4 standards

[10.2] Bootable ATA controller supports El Torito No Emulation mode

10.3 [REDUNDANT] Option ROMs support Int 13h Extensions

[10.4] If implemented, dual ATA adapters use single FIFO with asynchronous access or dual FIFOs and channels

[10.6] [REDUNDANT] System BIOS supports ARMD

[10.7] Controller supports Ultra DMA (ATA/33 or higher)

[10.8] Controller and peripheral connections include Pin 1 cable designation with keyed and shrouded connectors

[10.11] BIOS enumeration of all ATAPI devices complies with ATA/ATAPI-4

[10.13] [REDUNDANT] Each controller has a Plug and Play device ID

[10.14] [REDUNDANT] Dynamic resource configuration is supported for all controllers

[10.15] [REDUNDANT] Resource configuration meets bus requirements

[10.17] ATA Channel complies with device class power management reference specification

[9.1] All PCI components comply with PCI 2.2

[9.3] PCI-to-PCI bridges comply with the PCI to PCI Bridge Specification, Revision 1.1

[9.4] System provides 3.3 V to all PCI connectors

[9.5] PCI add-on devices support 3.3 V signaling

[9.6] System-board bus complies with PCI 2.2

[9.9] All PCI devices complete memory write transaction (as a target) within specified times

[9.11] PCI Device IDs include Subsystem IDs

[9.13] PCI interrupt routing is supported using ACPI

[9.14] BIOS does not configure I/O systems to share PCI interrupts

[9.15] BIOS configures boot device IRQ and writes to the interrupt line register

[9.16] System that supports hot plugging for any PCI device uses ACPI-based methods

[NEW] System that supports PCI-X complies with PCI-X 1.0

[9.17] All PCI components comply with PCI Bus Power Management Interface specification

[9.18] System provides support for 3.3 Vaux

[9.19] PCI bus power states are correctly implemented

[9.20] PCI-based modem and network adapters support wake-up

[6.26] System supports PCI docking through a bridge connector

[13.8] [REDUNDANT] All devices meet PC 2001 general device requirements